7/PRTS

VISION ARCHITECTURE TO DESCRIBE FEATURES OF PERSONS

INS B1

Field of the Invention

The present invention relates to machine vision systems, and more particularly, to vision systems configured to describe and recognize person's head and hand features.

Background of the Invention

Existing vision systems often implement computationally intensive processes for locating a person in an image frame and determining fiducial points on the person's head or hands. Such computationally intensive techniques are not amenable to economical and cost effective real time recognition systems.

Accordingly, there exists a significant need for vision systems that implement efficient recognition system processes. The present invention satisfies this need.

Summary of the Invention

The present invention is embodied in a method, and related apparatus, to determine the state of a person in an image, comprising defining a region of interest including a predetermined feature of the person and analyzing the region of interest using graph matching.

In more detailed features of invention, the step of defining a region of interest includes the use of early vision cues. The early vision clues may include at least one of stereovision, motion, color, convexity, topology, and structure. The stereovision may be used to produce disparity histograms and silhouette images.

In other more detailed features of the invention, the step of defining the region of interest may include background suppression. Further, the current state of a person's face may be described by node positions and outputs of Gabor kernals.

Other features and advantages of the present invention should be apparent from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the invention.

Brief Description of the Drawings

Fig. 1 is a block diagram of a machine vision apparatus and process, according to the invention.

Fig. 2 is schematic diagram related to a convex detector, according to the invention.

Fig. 3 includes schematic diagrams showing an original image and resulting Gabor wavelets, jets, graphs and bunch graphs.

Fig. 4 is adjacent facial images showing Gabor kernels for finding corresponding facial features.

Fig. 5 is a schematic diagram indicating finer analysis for eye and mouth regions, according to the invention.

Fig. 6 is a series of facial images tracking facial features over a sequence of 25 frames.

Fig. 7 is a face image with an overlying graph that specializes on specific poses.

Fig. 8 is a face image with background suppression.

Detailed Description of the Preferred Embodiments

With reference to the drawings, the invention is embodied in a machine vision apparatus 10, and related method, that allows the description of persons appearing in video images. It integrates vision routines that detect heads and hands with modules that perform pattern recognition to analyze the heads, faces and hands in fine detail. Head and hand detection makes use of a broad integration of different visual pathways such as motion, color and stereo vision as well as modules that extract topological and structural cues. Pattern recognition for fine analysis makes use of the technique known as elastic bunch graph matching.

The integration of early vision routines that detect heads and hands with routines that account for detailed pattern analysis is needed because pattern analysis alone is not robust enough against variations in orientation, pose, illumination or occlusions. Making pattern recognition robust against such variations is currently only possible at high computational costs.

Pattern analysis consists of several steps. First, it aims at finding fiducial points in the image that correspond to such features as center of an eye or fingertip. To this end a coarse to fine approach is adopted that first locates the fiducial points roughly and then, in subsequent refinement steps, with a higher level of accuracy. Once the facial features have been found, a tracking process keeps track of the feature positions. Finally, the features extracted at fiducial points are compared against features extracted at the corresponding locations in gallery images. This division of the pattern analysis process is helpful because the initial landmark finding is very time consuming and on typical cost effective hardware cannot be performed at the frame rate. Tracking works much faster and can run faster than frame rate. Thus, while initial landmark finding takes place, the buffer is filled with new incoming images. When tracking starts, the system catches up and the buffer is cleared.

With reference now to Fig. 1, to describe a person's face using a captured image it is first necessary to roughly locate the head in the scene. This is achieved with the head detection and tracking modules 12. A preselector module 14 selects the most suitable views for further analysis and refines the head detection such as to center and scale the head more properly. Then a landmark finding process 16 is used to detect the individual facial features. A facial feature tracking module 18 can be added to keep track of the landmarks found. The features extracted at the landmarks can then be compared against galleries in comparison processes 20. The landmark finding module is generally required while the other modules may be added according to the needs of the application. After an initial landmark finds facial feature tracking operates directly on the incoming images.

To choose the appropriate method for head detection and tracking, one has to take a closer look at the image material at hand. We discriminate between three cases that frequently occur in practice: a single image, a monocular image stream and a stream of stereo images. These three conditions offer increasingly richer information to be exploited. In case of single images, face detection has to be based on pattern analysis. Image streams additionally allow for optical flow methods to be employed, while stereo camera systems also offer the possibility to infer range. Whatever is possible to achieve in a more difficult paradigm with less information available is of course also possible in conditions where more information is available.

In a situation where only a single image is available we have two possible pathways at hand in order to detect the face. In case that the face covers a significant portion (say, at least 10%) of the image, we can use the elastic bunch graph matching in order to find a face. Elastic bunch graph matching is discussed in Wiskott et al., "Face Recognition by Elastic Bunch Graph Matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7): 775-779 (1997), which is incorporated herein by reference. If the faces are smaller, the most reliable method we know of is the neural network based face detector developed by Rowley et al., 1998. If color information is available, the skin color detection can be used to increase reliability of the face detection. A skin color detector can be based on a look-up table that contains possible skin colors. Confidence values indicating the reliability of face detection that generated during bunch graph matching or within the neural network are increased for skin-colored image regions.

Image streams (monocular) allow for the analysis of image motion. Exploiting image motion works particularly well for single persons moving in front of a stationary background. In other situations characterized by movements of multiple persons or strong background motion, the motion cue is less useful and one is essentially forced to return to the methods applicable for single images

Forming difference images is the simplest method to tell which regions in an image have been moving. Optical flow methods, as described in Jepson and Fleet, Measurement of Image Velocity, 1992, provide an alternative and more reliable means to determine which image regions change but are computationally more demanding.

In detecting motion of heads or hands we can exploit the heuristic that they often belong to convex regions within a motion silhouette. Several methods for determining convex regions of a noisy motion silhouette are known (see e.g. Turk et al., "Eigenfaces for Recognition", Journal of cognitive Neuroscience, Vol. 3, No. 1, P. 71, 1991, which is incorporated herein by reference). Under the assumption of a single person in an upright position and static backgrounds, the following method works well to locate a head. The binary motion map that indicates which regions were changing, is treated with a clustering algorithm that groups moving regions. Then we determine the top of the highest cluster within the image that exceeds a minimal threshold size and measures the diameter of the cluster at a fixed distance below the top. The top of the cluster is treated as the upper tip of the head and the diameter of the cluster is regarded as the diameter of the head image.

As shown in Fig. 2, a convex detector checks whether a pixel that belongs to a motion silhouette has neighbors that fall into a certain allowed region on the circumference. The connected allowed region can be located in any part of the circumference. The output of the convex detector is binary.

Skin color within an image is again an important indicator for the presence of heads and hands. Again it is often helpful to employ a convex detector similar to the one described above to find convex regions in skin color maps which have an even higher chance of showing a head or a hand.

Also in case that head detection is primarily based on motion and color cues, it is beneficial to employ a neural network face detector to verify the hypothesis arising from the exploitation of these cues.

Very reliable and fast face detection is possible if a stream of stereo images is available. This is because stereo allows discriminating between foreground and background and it allows for determining the image size of objects of a known size. The latter is the case for heads and hands. Knowing the expected image size of a head is of course very helpful in the detection process.

To perform a reliable stereo analysis we first determine the image regions subject to image motion as well as the skin color regions in case that the color is available. A stereo algorithm then separately determines the stereo disparities of those pixels that have changed or exhibit skin color. The fact that the stereo algorithm only attempts to match moving pixels with moving pixels and skin colored pixels with skin colored pixels reduces the search space for the stereo matching process. This has as an effect that computation time as well as the number of erroneous matches is reduced. We exploit the disparity information by using disparity histograms. A disparity histogram shows the number of pixels that have a certain disparity against this disparity. Then, image regions confined to a certain disparity interval are selected by inspecting the local maxima of the disparity histogram. Sets of pixels that have changed or have skin color and belong to a neighborhood of a local maxima are referred to as motion or color silhouettes. Silhouettes are binary images.

Again it is often useful to look for convex regions within the silhouettes. To this end the convex detector described in Fig. 2 is suitable.

Motion silhouettes, skin color silhouettes, outputs of the convex detectors applied to the motion silhouettes and outputs of the convex detectors applied to the skin color silhouettes, provide four different evidence maps. An evidence map is a scalar function over the image domain that indicates the evidence that a certain pixel belongs to a face or a hand. Each of the aforementioned four evidence maps is binary valued. The available evidence maps are linearly superimposed for a given disparity and checked for local maxima. Local maxima indicate candidate positions where heads or hands might be found. The expected diameter of a head can be inferred from the local maximum in the disparity map that gave rise to the evidence map at hand. Head detection as described here performs well even in the presence of strong background motion.

In case of image sequences it is often interesting to concatenate the individual position to trajectories for head tracking. Since motion analysis is often an essential step in head detection, it is particularly important to account for periods when the person remains still.

Head tracking consists of the following steps (for details see Rehberg, Master's Thesis, University of Bochum, Germany, Institute for Neural Informatics, 1997, which is incorporated herein by reference). In a preliminary step, a thinning takes place that represents position estimates coming from head detection and which are close to each other by a single representative estimate only. Second, it is checked whether the new position estimate belongs to an already existing trajectory. This is achieved by assuming spatio-temporal continuity. For every position estimate found for the frame acquired at time t, the algorithm looks for the closest head position estimate that was determined for the previous frame at time t-1 and connects it. If no estimate can be found that is sufficiently close, it is assumed that a new head appeared. To connect individual estimates to trajectories we only work with image coordinates.

Every trajectory has an assigned confidence value that is updated using a leaky integrator. If the confidence value falls below a fixed threshold, the trajectory is deleted. To stabilize trajectory creation and deletion we employ a hysteresis mechanism in the way that in order to initiate a trajectory, a higher confidence value has to be reached for the trajectory deletion.

In order to describe the remaining modules, it is helpful to have a short look into the method of elastic graph matching. For details, please refer to Wiskott et al. 1997, supra. As a basic visual feature we use a local image descriptor represented by a jet. Each component of a jet is the filter response of a Gabor wavelet extracted at a point (x, y) of the image. A Gabor wavelet consists of a two-dimensional complex wave field modulated by a gaussian envelope. We typically use wavelets of five different frequencies and eight different orientations. Thus a jet may contain 40 complex values. It describes the area surrounding the position (x,y). A set of jets taken at different positions form a model graph representing the face in the image. The nodes of the graph are indexed and interconnected. Nodes and edges define the graph topology. Graphs with equal geometry can be compared. The normalized dot product of the absolute components of two jets defines the jet similarity. This value is independent against illumination and contrast changes. To compute the similarity between two graphs, we take the sum over similarities of corresponding jets between the graphs. Gabor wavelets, jets, graphs and bunch graphs are shown in Fig. 3.

In order to find a face in the image, a graph is moved and scaled over the image until we find a place is found where it matches best (the graph jets are most similar to jets extracted from the image at positions of the nodes). Since face features differ from face to face, try to make the graph more general for the task: assign to each node jets of the corresponding landmark taken from 10 to 100 individual faces. This enhanced model graph structure is called bunch graph. Fig. 4 shows a technique for finding of corresponding features

The preselector takes as an input a series of face candidates that belong to the same trajectory as determined by head tracking. The preselecting process is particularly useful in case person spotting is not based on facial feature tracking. The preselecting module crops the head region out of the incoming image. It uses elastic graph matching with a small bunch graph in order to find a face in the image sequence. This bunch graph typically consists of about 10 faces. The jets also tend to have less orientations and frequencies. A typical preselector jet contains 12 values (4 wave field orientations and 3 frequencies). The similarity achieved during face finding acts as a measure of suitability of the face for further processing. The image of a sequence leading to the highest similarity is selected for landmark finding. It is called probe image. After matching, the face position is derived from the center of gravity of all node positions. The mean euclidean distance of all nodes from the center of gravity defines a canonical graph size value, which is used for face size estimation. These two measures are more accurate than the head position and size estimation of head tracker. Preselector crops and rescales the face accordingly and sends the resulting image to the landmark finder.

Landmark finding is a process that determines the image locations of facial features. Two different approaches. are employed. One approach makes use of a family of two-dimensional bunch graphs defined in the image plane (Wiskott et al 1997). The different graphs within one family account for different poses and scales. If interested in one particular pose, for instance the frontal pose, the family might consist of only one single bunch graph. The second approach uses only one graph defined in 3D space. For instance one uses a model of an average head in order to define the 3D graph for a head. As in the 2D approach the nodes are located at the fiducial points on the head surface. Projections of the 3D graph are then used in the matching process. An important generalization of the 2D approach is that every node has an attached parameterized family of bunch jets. The parameters typically consist of three angles describing the pose, and scale parameters.

The matching process that involves these graphs is often formulated as a coarse to fine approach that first utilizes graphs with fewer nodes and kernels and in subsequent steps more dense graphs. This coarse to fine strategy is applicable in the 2D as well as in the 3D domain. A particular version of a coarse to fine approach is suitable if one is interested in high precision localization of the feature points in certain areas of the face. In this case it saves on the computational effort to adopt a hierarchical approach in which landmark finding is first performed on a coarser resolution, and subsequently the adapted graphs are checked at a higher resolution to analyze certain regions in finer detail. For example, as shown in Fig. 5, after the eye and mouth regions have been found, a finer analysis is performed at a higher resolution.

Once the landmarks are found, a process that keeps track of the facial landmarks sets in. The basic method is described in Maurer and von der Malsburg, "Tracking and Learning Graphs and Pose on Image Sequences of Faces", Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition, IEEE Comp. Soc. Press, pp. 176-181, 1997, which is incorporated herein by reference. To find the corresponding node positions in the new frame, only the jets extracted in the actual frame are used, i.e., the system has one single graph in memory which is matched on a new frame, then replaced, and so on. This way we get a general tracking device, which can be further optimized for different applications by including additional constraints.

To compute the displacement of a single node between two consecutive frames, a method is used, developed for disparity estimation in stereo images, based on Jepson and Fleet, 1992, supra, and Theimer and Mallot 1994, "Phase-based binocular vergence control and depth reconstruction using active vision", CVGIP: Image Understanding, vol. 60, pp. 343-358, Nov. 1994, which is incorporated herein by reference. The strong variation of the phases of the complex filter responses is used explicitly to compute the displacement with subpixel accuracy (Wiskott 1997, supra). By writing the response J to the jth Gabor filter in terms of amplitude $a_j$and phase j a similarity function can be defined as

$$S(J,J',d) = \frac{\sum_j a_j a'_{j'} \cos(\phi_j - \phi_{j'} - d \cdot k_j)}{\sqrt{\sum_j a_j^2 \sum_{j'} a'^2_{j'}}}$$

Let J be the jet at some position x n frame n, and J' the jet at the same position x in the next frame n+1, the displacement d of the corresponding point can be found by maximizing the similarity S with respect to d, the $k_j$being the wavevectors associated with the filter generating J_j. Because the estimation of d is only precise for small displacements, i.e., large overlap of the Gabor jets, large displacement vectors are treated as a first estimate only, and the process is repeated. This way displacements up to half the wavelength of the kernel with the lowest frequency used can be computed (see Wiskott 1995 for details). For our Gabor kernels the maximal displacement is 6 to 7 pixels. As already mentioned in the introduction, a much larger range would help only in the special case of a purely translational movement, in all other cases larger displacements are associated with greater changes in the image, and then the corresponding node position might not be found anyway. But if fast frontoparallel motion should cause problems, this could be easily remedied by including the assumption of continuity in the motion, i.e., by starting the computation of d_{n+1} not at x_n, but at $(x_n + d_n)$.

Thus, for all nodes of the graph in frame n, the displacement vectors with respect to frame n+1 are computed, then a graph is created with its nodes at these new corresponding positions in the new frame, and all stored jets (which had been extracted in frame n, are replaced by those extracted at the corresponding node positions in frame n+1. But here we have a problem: Although the displacements have been determined as floats, the jets can be extracted at (integer) pixel positions only, resulting in a systematic rounding error. To compensate for this subpixel error Δd, the phases of the complex Gabor filter responses must be shifted according to

$$\Delta\phi_j = \Delta d \cdot k_j$$

then they will look as if they were extracted at the correct subpixel position. This way the Gabor jets can be tracked with subpixel accuracy without any further bookkeeping of rounding errors. This is an additional advantage of using Gabor jets in image processing; subpixel accuracy is a more difficult problem in most other image processing methods. Fig 6. Facial features tracked over a sequence of 25 frames

The tracking of facial features alone is unstable since this module tracks image structures from frame to frame without "knowledge" about the appearance of typical facial features. Therefore it is necessary to introduce a correction mechanism that uses face knowledge. To this end it is useful to employ bunch jets extracted from a number of stored example faces in local searches that try to correct erroneous jet positions. If too many nodes are off, it is helpful to reinitialize the tracking by a landmark finding process as described before. Since the tracking provides some estimate about the current head pose an appropriate bunch graph for the given pose can be selected and matched as shown in Fig. 7. In the case a 3D graph is used for landmark finding the appropriate scale and orientation can be determined from the positions of the tracked nodes. Here, in contrast to the initial landmark finding it is not necessary to check for multiple poses simultaneously. Therefore the re-initialization process can be much faster.

Once the facial feature positions are known for a given frame, the jets extracted at these positions can be compared with the jets extracted from stored gallery images. Either complete graphs are compared, as it is the case for face recognition applications, or just partial graphs or even individual nodes are. For instance in order to determine the degree to which an eye is closed, it is appropriate to compare only the jets extracted from the eye region.

Before the jets are extracted for the actual comparison, a number of image normalizations are applied. One such normalization is called background suppression. The influence of the background on probe images needs to be suppressed because different backgrounds between probe and gallery images lower similarities and frequently leads to misclassifications. Therefore, the nodes and edges surrounding the face are taken as face boundaries. Background pixels get smoothly toned down when deviating from the face. Each pixel value outside of the head is modified as follows:

$$p_{new} = p_{old} \cdot \lambda + c \cdot (1 - \lambda)$$

where

$$\lambda = \exp(-\frac{d}{d_0})$$

and c is a constant background gray value that represents the euclidean distance of the pixel position from the closest edge of the graph. d_0 is a constant tone down value. Of course, other functional dependencies between pixel value and distance from the graph boundaries are possible. As shown in Figure 8, the automatic background suppression drags the gray value smoothly to the constant when deviating from the closest edge. This method still leaves a background region surrounding the face visible, but it avoids strong disturbing edges in the image, which would occur if we simply filled up this region with a constant gray value.

The described system can be readily adapted to serve in various applications. The following ones seem particularly interesting.

For example, the above system can be adapted to perform person spotting from live video. The comparison is then against a gallery with stored facial images, which are then compared against incoming probe images in order to recognize persons. Two versions of the person spotting system are possible. One version makes use of the preselector module to select a few suitable images for recognition out of a series of face images belonging to the same trajectory. The other version does not use the preselector but instead uses the facial feature tracking to generate a sequence of matched graphs which are then compared against the gallery.

The above system can be adapted to perform automated caricaturing. Systems that attempt to generate caricatures often use on a number of templates for different facial features in order to assemble the caricature. This process needs of course the locations of the different facial features. In addition they need to classify the different facial features. This classification can be based on the location of various fiducial points and on jet comparisons. In the latter case it is necessary to provide example galleries for the different facial features that contain prototypes of the different classes of interest.

Additionally, the information that becomes available during facial feature tracking. i.e. the node positions as well as the information contained in the jets, can be used to animate a graphical head model. Also, facial feature detection and tracking as described above is useful in image encoding.

Further, the information contained in the partial graphs covering the eyes can be used to obtain information that is useful in detecting drowsiness.

Although the foregoing discloses the preferred embodiments of the present invention, it is understood that those skilled in the art may make various changes to the preferred embodiments without departing from the scope of the invention. The invention is defined only the following claims.